Date: Jan. 18, 2000 Express Mail Label No. EL290726531US

Inventors: Gary Lewis

Attorney's Docket No.: 2386.1012000 (CISCO # 46741)

VOICE QUALITY IMPROVEMENT FOR VOIP CONNECTIONS ON LOW LOSS NETWORKS

BACKGROUND OF THE INVENTION

Real-time audio, such as a telephone conversation may be transmitted over a data network, such as the Internet. The audio transmitted during the telephone conversation includes desired audio (spoken words) and undesired audio (background noise), such as the sound of the air conditioner. While words are being spoken the transmitted audio contains both spoken words and background noise. While words are not being spoken, the transmitted audio contains only background noise.

To transmit real-time audio over the data network, an audio packet transmitter in the source stores the audio in the payload of one or more data packets and transmits the data packet over the data network. Each data packet includes a destination address in a header included in the data packet.

Unlike a telephone network in which there is a dedicated connection between the source and the destination, each data packet may travel on a different path from the source to the destination in a data network and some data packets may travel faster than others. Thus, data packets transmitted over the data network may arrive out of order at the receiver.

To compensate for these path differences, an audio packet receiver in the destination stores the received data packets in a jitter buffer and forwards the stored audio to the listener at the rate at which it was generated in the audio packet transmitter

in the source. Jitter buffer latency is the period of time that the received data packet is stored in the jitter buffer being forwarded to the listener. Thus, the jitter buffer latency is the delay after which the receiver forwards the received data packet to the listener. The jitter buffer latency is dependent on the size of the data packet being transmitted and the slowest path between the source and the destination. However, if data packets are not being received on the slowest path, the jitter buffer latency may be reduced.

Thus, in a low loss network, the inter-packet arrival time is monitored and the jitter buffer latency is modified dependent on the inter-packet arrival time, in order to minimize the delay. This modification of the jitter buffer latency is performed while no spoken words are being transmitted so as to minimize the loss of spoken words.

One standard protocol for packetizing real-time audio for transmission over a data network is the Real-Time Transport Protocol ("RTP") (Request for Comments ("RFC") 1889, Jan 1996) at http:// www.ietf.org/rfc/rfc1889.txt. The RTP provides a method for a transmitter to detect the start of a period in which the audio does not contain spoken words. The period in which the audio does not contain spoken words is sometimes called "a period of silence" even though it is not true silence because the audio contains background noise. Upon detecting a period of silence the transmitter may either transmit no data packets during the period of silence or transmit non-speech audio (background noise with no spoken words). By transmitting no data packets during a period of silence, the audio packet receiver may adjust the jitter buffer latency while no data packets are being transmitted and the number of spoken words lost is minimized.

Thus, to minimize the number of spoken words lost while the jitter buffer latency is modified, the transmitter does not transmit non-speech audio packets during the period of silence. During the period of silence, the receiver generates comfort noise to reconstruct background noise for the listener. The receiver forwards the comfort noise to the listener. Comfort noise is generated and forwarded to reassure the listener that the telephone conversation has not ended. The comfort noise reduces the quality of the real-time audio because the listener hearing comfort noise during a period of silence in a telephone conversation instead of background noise receives a negative indication or impression that

the telephone conversation is being transmitted over a data network instead of a telephone network.

SUMMARY OF THE INVENTION

A non-speech identifier is stored in a data packet if the data packet contains non-speech audio. The non-speech audio in the data packet is identified dependent on the state of the non-speech identifier. Upon detection of a data packet containing non-speech audio; jitter buffer latency modification is enabled. Upon detection of a data packet containing speech audio; jitter buffer latency modification is disabled.

The non-speech identifier may be a one bit field stored in the header of the data packet. The header may be the Real-time Transport Protocol header. The one bit field is set to a first of two states if the type of audio stored in the data packet's payload is non-speech audio and set to a second state if the type of audio stored in the data packet's payload is speech audio.

BRIEF DESCRIPTION OF THE DRAWINGS

The foregoing and other objects, features and advantages of the invention will be apparent from the following more particular description of preferred embodiments of the invention, as illustrated in the accompanying drawings in which like reference characters refer to the same parts throughout the different views. The drawings are not necessarily to scale, emphasis instead being placed upon illustrating the principles of the invention.

Fig. 1 illustrates a data network including an Internet access server for transmitting audio on the Internet according to the principles of the present invention;

Fig. 2 illustrates any one of the Internet access servers shown in Fig. 1;

Fig. 3 illustrates a prior art Real-time Transport Protocol ("RTP") data packet including an RTP header transmitted by the Internet access server shown in Fig. 2;

Fig. 4 illustrates the format of the prior art RTP header shown in Fig. 3;

Fig. 5 is a flowchart illustrating the steps implemented in the add header routine in the audio packet transmitter shown in Fig. 2 for storing a non-speech identifier in the data packet's header;

Fig. 6 is a flowchart illustrating the steps implemented in the remove header routine in the audio packet receiver shown in Fig. 2 for detecting non-speech data in a data packet's's payload dependent on the state of the non-speech identifier stored in the data packet's header.

DETAILED DESCRIPTION OF THE INVENTION

Fig. 1 illustrates a data network 100 including Internet access servers 104a-c transmitting real-time audio on the Internet according to the present invention. The real-time audio input to the Internet access servers 104a-c may originate in telephones 112a-b connected to a Public Switched Telephone Network ("PSTNs") 106 or telephones 112c-d connected to Private Branch Exchange ("PBXs") 110, or a telephone 112e directly connected to an Internet access server 104c. The Internet access server 104 may be a personal computer. The Internet access servers 104a-c packetize the real-time audio by portioning the real-time audio into payloads and storing the payload and an associated header in a data packet. The Internet access server 104a-c transmits the data packet to another Internet access server 104. Upon receiving the data packet, the Internet access server 104 removes the header and forwards the real-time audio stored in the payload of the data packet to a destination. The destination is dependent on the destination address included in the header of the data packet. A non-speech identifier stored in the header of the data packet identifies the type of audio stored in the payload. The type of audio stored in the payload is either speech audio or non-speech audio. If the type of audio is non-speech audio the audio stored in the payload is background noise. If the type of audio is speech audio the audio stored in the payload is spoken words and background noise.

For example, to transmit audio from telephone 112a to telephone 112e, audio originating in telephone 112a is forwarded to Internet access server 104a through the PSTN 106. Transmitting Internet access server 104a samples the audio and divides the sampled

audio into payloads. Internet access server 104a generates a network data packet by adding a network header to the payload. The header includes a destination address for the data packet and a non-speech identifier identifying the type of audio stored in the payload as speech audio or non-speech audio.

The data packet is forwarded on the Internet dependent on the destination address included in the header. Receiving Internet access server 104c determines from the destination address included in the header of the data packet that the destination address is telephone 112e, removes the header from the data packet and forwards the audio stored in the payload to telephone 112e.

To transmit audio from telephone 112e to telephone 112a, Internet access server 104c packetizes the audio forwarded from telephone 112e and Internet access server 104a de-packetizes the data packets and forwards the audio to telephone 112a through the PSTN 106.

Thus, to transmit audio on the Internet, the audio is packetized by an Internet access server 104a-c, forwarded on the Internet and de-packetized by the Internet access server 104a-c after it is received from the Internet.

Fig. 2 illustrates Internet access server 104a shown in Fig. 1. The Internet access server includes a Coder/Decoder ("CODEC") 200, an audio packet transmitter 220, an audio packet receiver 222 and a data network controller 210. The audio packet transmitter 220 includes an add header routine 202 and a speech detection module 212. The audio packet receiver 222 includes a remove header routine 206 and jitter buffer logic 204.

The CODEC 200 samples the audio input signal received on voice port 114. For example, if the CODEC implements the G.711 protocol the audio input is sampled 8,000 times per second. The sampled audio signal is converted into a digital format and encoded into frames.

The add header routine 202 in the audio packet transmitter 220 generates a network data packet by adding data network protocol headers to the encoded frame or payload received from the CODEC 200. The data packet is forwarded to a data network controller 210, such as an Ethernet controller. The data network controller 210 transmits the data packet on the Internet through the data network interface 116. The speech detection module

212 in the audio packet transmitter 220 determines whether the type of audio stored in the encoded frame is speech audio or non-speech audio and forwards a speech identifier 214 to the add header routine 202. The add header routine 202 sets the state of a non-speech identifier in the header of the data packet dependent on the state of the speech identifier 214 forwarded from the speech detection module 212. The audio packet transmitter 220 forwards the data packet to the data network controller 210.

Fig. 3 illustrates a data packet 304 including prior art data network headers 302a-c which may be added to the encoded frame received from the CODEC 200 by the add header routine 202 (Fig. 2). The data packet 304 includes a payload 300. The audio forwarded in the encoded frame is stored in the payload 300 of the data packet 304. The headers 302a-c include an Internet Protocol ("IP") header 302a, a User Datagram Protocol ("UDP") header 302b and a Real-time Transport Protocol ("RTP") header 302c.

Fig. 4 illustrates the format of the prior art RTP header 302c shown in Fig. 3. The RTP header 302c includes the following fields: version 400, padding 402, extension 404, contributing source count 406, marker 408, payload type 410, sequence number 412, timestamp 414, synchronization source identifier 416 and contributing source identifier 418.

The version 400 field identifies the version of the RTP header 302c. The version shown in Fig. 4 is version 2. The padding 402 field indicates whether the payload 300 (Fig. 3) includes padding octets. The extension 404 field indicates whether the RTP header 302c includes an extension header. The marker 408 field is user definable. In the embodiment shown, the marker 408 field is one bit wide. The use of the marker 408 field is described in later in conjunction with Figs. 2, 5 and 6.

The payload type 410 field identifies the format of the payload 300 (Fig. 3). The sequence number 412 field increments by one for each RTP data packet transmitted, it may be used by the receiver to detect lost data packets and to restore data packet sequence if data packets arrive out of order. The timestamp 414 field stores a timestamp dependent on the time the payload 300 (Fig. 3) was sampled. The synchronization source identifier 416 field identifies the synchronization source. The contributing source identifier 418 field identifies the contributing sources for the payload 300 (Fig. 3).

Returning to Fig. 2 the add header routine 202 generates a data packet 304 (Fig. 3) by adding an IP header 302a (Fig. 3), a UDP header 302b (Fig. 3) and an RTP header 302c to the payload 300 (Fig. 3) forwarded from the CODEC 200 on data-out 216.

The add header routine 202 stores a non-speech identifier, identifying the type of audio stored in the payload 300 (Fig. 3) of the data packet 304 (Fig. 3). The non-speech identifier is stored in the marker 408 (Fig. 4) field in the prior art RTP header 302c (Fig. 3). The marker 408 (Fig. 4) field is a one bit wide field. The marker 408 (Fig. 4) field is set '1' if the type of audio stored in the payload 300 (Fig. 3) is non-speech audio and set '0' if the type of audio stored in the payload 300 (Fig. 3) is speech audio.

Fig. 5 is a flowchart illustrating the steps in the add header routine 202 (Fig. 2) for storing a non-speech identifier identifying the type of audio stored in the payload 300 (Fig. 3) in the data packet's RTP header 302c (Fig. 3) before the data packet 304 (Fig. 3) is transmitted on the data network.

At step 500, the add header routine 202 (Fig. 2) determines if the audio stored in the payload 300 (Fig. 3) of the data packet is speech audio dependent on the state of the speech identifier 214 (Fig. 2) forwarded from the speech detection module 212 (Fig. 2). If the speech identifier 214 (Fig. 2) indicates that the type of audio stored in the payload 300 (Fig. 3) is speech, audio processing continues with step 502. If not, processing continues with step 504.

At step 502, the marker 408 (Fig. 4) field in the RTP header 302c (Fig. 4) is set to '0' identifying the type of audio stored in the payload 300 (Fig. 3) as speech audio.

At step 504, the marker 408 (Fig. 4) field in the RTP header 302c (Fig. 4) is set to '1' identifying the type of audio stored in the payload 300 (Fig. 3) as non-speech audio.

Fig. 6 is a flowchart illustrating the steps in the remove header routine 206 (Fig. 2) in the audio packet receiver 222 (Fig. 2) for detecting the type of audio stored in the payload 300 (Fig. 3) of the received data packet 304 (Fig. 3).

At step 600, the remove header routine 206 (Fig. 2) examines the state of the marker 408 (Fig. 4) field stored in the RTP header 302c (Fig. 4) in the received data packet 304 (Fig. 4).

If the marker 408 (Fig. 4) field is set to '1', processing continues with step 602. If not, processing continues with step 604.

At step 604, the remove header routine 206 (Fig. 2) disables modify_latency_en 208 upon detecting that the type of audio stored in the payload 300 (Fig. 3) of the received data packet's payload is speech audio. The jitter buffer logic 204 (Fig. 2) may not modify the jitter buffer latency which modify_latency_en 208 (Fig. 2) is disabled.

At step 602, the remove header routine 206 (Fig. 2) enables modify_latency_en 208 (Fig. 2) upon detecting that the type of audio stored in the received data packet's payload 300 (Fig. 3) is non-speech audio. While modify-latency_en 208 is enabled, the jitter buffer logic 204 (Fig. 2) may modify the jitter buffer latency. Modifying the jitter buffer latency while the payload 300 (Fig. 3) contains non-speech audio minimizes the loss of spoken words.

Returning to Fig. 2, after the remove header routine 206 has removed the headers 302 (Fig 3) from the received data packet 304 (Fig. 3), the payload 300 (Fig. 3) is stored in a jitter buffer (not shown) in the jitter buffer logic 204. The jitter buffer logic 204 forwards the stored payload 300 (Fig. 3) -- the encoded frames -- to the CODEC 200 on data-in 218 dependent on the jitter buffer latency. The CODEC 200 converts the received encoded frame to a format to be transmitted on audio signal 114 (Fig. 1) to the PSTN 106 (Fig. 1). .

The invention transmits non-speech audio in the payload of a data packet; that is, background noise with no spoken words. The type of audio stored in the payload of the data packet is identified as by storing a non-speech identifier in the RTP header of the data packet. The standard RTP header is used to identify the type of audio stored in the data packet's payload. Upon detection of non-speech audio in the payload of the data packet, the receiver of the data packet may modify the latency of the jitter buffer without loss of spoken words.

It will be apparent to those of ordinary skill in the art that methods involved in the present invention may be embodied in a computer program product that includes a computer usable medium. For example, such a computer usable medium can consist of a read only

memory device, such as a hard drive device or a computer diskette, having computer readable program code stored thereon.

While this invention has been particularly shown and described with references to preferred embodiments thereof, it will be understood by those skilled in the art that various changes in form and details may be made therein without departing from the scope of the invention encompassed by the appended claims.